STATUS OF THE CLAIMS

This listing shows the current state of the claims in the application.

Listing of Claims:

1-20. (canceled)

21. (currently amended) An apparatus comprising:

an integer pipeline to process integer instructions;

a floating point pipeline to process short latency and long latency floating point instructions in which both short and long latency floating point instructions generate exceptions, wherein the floating point pipeline has a greater number of pipeline stages to process floating point instructions than a number of pipeline stages to process integer instructions in the integer pipeline; and

a control circuit coupled to the integer and floating point pipelines to inhibit co-issuance of an integer instruction to the integer pipeline when the integer instruction is subsequent to a first floating-point instruction in program order, until the first floating point-instruction reaches a stage in the floating-point pipeline where exceptions are to be generated to ensure that the integer instruction does not graduate from the integer pipeline prior to exception determination for the first floating point instruction in the floating-point pipeline, the control circuit to also inhibit co-issuance of a second floating-point instruction that follows the first floating-point instruction in program order, if the first floating-point instruction is ~~not a short~~ a long latency floating-point instruction, to ensure that the second floating-point instruction does not graduate prior to the exception determination for the first floating point instruction, but not to inhibit the second floating-point instruction from co-issuance if the first floating-point instruction is a short latency floating-point instruction, since the second floating-point instruction will not graduate prior to the exception determination for the first floating-point instruction.

22. (currently amended) The apparatus of claim 21 wherein the first floating-point instruction is a multiply-add instruction ~~that has a longer latency period than the short~~

~~latency floating point instruction~~ that is one of the long latency floating point instructions.

23. (canceled)

24. (previously presented) The apparatus of claim 21 further comprising a load/store pipeline coupled to the control circuit to process load and store instructions, the load/store pipeline also to be inhibited for co-issuance of load or store instruction, which is subsequent to the first floating-point instruction in program order, until the first floating-point instruction reaches the stage in the floating-point pipeline where exceptions are to be generated.

25. (currently amended) A method comprising:

queuing a first floating-point instruction for issuance to a floating-point pipeline to process the first floating-point instruction, wherein the first floating point instruction is either a short latency floating point instruction or a long latency floating point instruction, in which both the short and long latency floating point instructions generate exceptions;

determining if exception handling for floating-point instructions is enabled;

inhibiting co-issuance of an integer instruction to an integer pipeline when the integer instruction is subsequent to the first floating-point instruction in program order and the exception handling is enabled, until the first floating point-instruction reaches a stage in the floating-point pipeline where exceptions are to be generated to ensure that the integer instruction does not graduate from the integer pipeline prior to exception determination for the first floating point instruction in the floating-point pipeline;

determining if the first floating-point instruction is ~~a~~ the short latency floating-point instruction or the long latency floating point instruction;

inhibiting co-issuance of a second floating-point instruction that follows the first floating-point instruction in program order and the exception handling is enabled, if the first floating-point instruction is ~~not a short~~ the long latency floating-point instruction, to ensure that the second floating-point instruction does not graduate prior to the exception determination for the first floating point instruction, but not to inhibit the second floating-

point instruction from co-issuance if the first floating-point instruction is ~~a~~ the short latency floating-point instruction, since the second floating-point instruction will not graduate prior to the exception determination for the first floating-point instruction.

26. (currently amended) The method of claim 25 wherein the ~~first~~ long latency floating-point instruction is a multiply-add instruction ~~that has a longer latency period than the short latency floating-point instruction~~.

27. (canceled)

28. (currently amended) The method of claim 25 further comprising inhibiting co-issuance of a load or store instruction to a load/store pipeline when the load or store instruction is subsequent to the first floating-point instruction in program order and the exception handling is enabled, until the first floating point-instruction reaches the exception determination stage in the floating-point pipeline.